# Knowledge Discovery in an Object-Oriented Oceanographic Database System

## Annual Report
**September 1, 1996 - August 31, 1997**

**ONR EPSCoR Grant N00014-96-1-1276**
**PR Number 96PR07924-00**

**Julia Hodges**
**Susan Bridges**

**Department of Computer Science**
**Mississippi State University**
**Box 9637**
**Mississippi State, MS 39762-9637**
**hodges@cs.msstate.edu**

### Abstract

The rate at which scientific data is collected today has overwhelmed the ability of scientists to store and analyze the data. Current research in knowledge discovery in databases is addressing this problem by developing techniques that can consider large quantities of data and automatically identify information that is of interest in a particular problem domain.

This report describes the results of the first year's efforts in the development of a knowledge discovery system for use by oceanographers at the Naval Oceanographic Office at the Stennis Space Center in the identification of certain oceanographic features. The system consists of two major components: an object-oriented oceanographic database that can support the retrieval of data along various parameters of interest (such as a certain geographic area or a certain date) and a discovery system that can identify the features of interest. During the first year of this project, we (in consultation with the scientists at the Stennis Space Center) named the identification of sand waves in acoustic imagery data as the first task to be addressed by our system.

## 1. Introduction

The ability in today's technology to collect extremely large quantities of scientific data has outpaced the ability of scientists to store and analyze this data. This is due both to a shortage of qualified analysts and to the overwhelming quantities of data being collected. Knowledge discovery in databases (KDD) is an active area of research that has developed in response to this problem. Fayyad, Piatetsky-Shapiro, and Smyth (1996) describe the problem as the need to develop "a new generation of techniques and tools with the ability to intelligently and automatically assist humans in analyzing the mountains of data for nuggets of useful information."

The goal of this project is to develop a knowledge discovery system consisting of an object-oriented oceanographic database and the tools need to support the automated extraction of information from the database. Such a system will aid oceanographers in the analysis of complex oceanographic data sets that are too large to be analyzed manually. This work involves two major efforts: the development of an object-oriented database capable of representing complex scientific data that cannot be supported by traditional relational database systems, and the development of knowledge discovery tools that aid oceanographers in their data analysis tasks. This work is being done in collaboration with scientists at the Naval Oceanographic Office at the Stennis Space Center. Although we have consulted with a number of scientists at NAVOCEANO at Stennis, our primary points of contact have been Dr. Martha Head, Supervisory Oceanographer, Modeling and Techniques Department, and Mr. Steve Lingsch, Geophysicist, Geophysical Techniques Department.

The scientists at NAVOCEANO are interested in a knowledge discovery system that can aid in the identification of certain oceanographic features. We are working with them to develop a prototype knowledge discovery system that uses acoustic imagery and other data to province the ocean floor. Currently the prototype system assists in the identification of provinces that contain sand waves, a task which the NAVOCEANO scientists chose as the initial task for the system.

## 2. The Oceanographic Database

We initially established the design and implementation of the object-oriented oceanographic database as the first major task in this project. The underlying reason for this decision was that the database could provide us with flexible access to the oceanographic data along various parameters of interest (such as a certain geographic area or a certain date). The database will include a variety of oceanographic data, such as satellite imagery, grid data (e.g., bathymetry data and model output), acoustic imagery, altimetry data, and data from a variety of sensors.

The size and complexity of some of this data, such as the acoustic imagery data, has prohibited its being stored in a relational database. For example, the representation of an analyzed acoustic image is a *complex object* - an object which has other objects as attributes. For purposes of textural analysis, an acoustic image is considered to consist of rectangular cells called *texels*, or texture elements (Reed and Hussong 1989). Each texel, in turn, consists of a number of *pixels* (picture elements). These texels are assigned to particular classes, and texels are grouped to together based on these classes to form provinces. Because of the complexity of the acoustic imagery data, NAVOCEANO currently stores only descriptive information about the data in a database, maintaining the imagery data itself in compressed form. Since we have developed for another project an object-oriented oceanographic database that contains many of the same types of data that the database for this project must support, we intend to use that database design as a starting point.

As we began the analysis of the requirements of this database system, we realized that we must have a better understanding of the data and the features of interest to the scientists at NAVOCEANO before we can identify (1) the appropriate metadata for describing the different types of data in the database and the different kinds of features that may be identified, and (2) the appropriate object design for the complex objects to be represented in the database. We experienced a delay in the arrival of ObjectStore, the object-oriented database management system that we are using for this project. We also experienced a delay in expected improvements in the computing facilities that we had expected to be available to us at the time this project began. (We have since acquired machines with more memory and faster processors in order to handle the large, complex data sets.) For these reasons, we modified our approach to the problem by focusing on the application of machine learning techniques to a manageable subset of the acoustic imagery data as our first task. We have only recently begun to identify the various parameters of interest and the metadata needed for the database. The design and implementation of the object-oriented oceanographic database is to be accomplished during the second year of this project.

## 3. The Knowledge Discovery System

The focus of most work in KDD has been the development of learning algorithms. Our experience, and that of others in the field, however, has shown that this is only one aspect of the technically challenging, multi-step, iterative process of knowledge discovery (Brachman and Anand 1996; Chen, Han, and Yu 1996). In their list of lessons learned in the application of AutoClass to real databases, the first item listed by Cheeseman and Stutz (1996) is:

> Data analysis/knowledge discovery is a process. Discovery of patterns in data is only the beginning of a cycle of interpretation followed by more testing.

In our work on knowledge discovery from oceanographic data, we have found that this process involves iteration over and within the following steps (at a minimum): preprocessing each type of data, extracting features from each type of processed data, selecting relevant features, applying one or more learning algorithms to the data, analyzing the results, and soliciting evaluations of the "interestingness" of the results from domain experts. We have found that many of these steps are very computationally expensive and that visualization of results and data at each step is critical.

The overall goal of the knowledge discovery system is to aid the scientists at NAVOCEANO in the analysis of large sets of complex oceanographic data. More specifically, the scientists at NAVOCEANO wish to have a system that can use acoustic imagery and other data to province the ocean floor. Geologists currently do this job manually. We wish to provide the geologists with a tool that identifies texture classes in the acoustic images, thus making it possible for them to province the ocean floor more efficiently. Figure 1 illustrates the knowledge discovery process that we are using to extract texture information from the acoustic images and to apply clustering algorithms to identify classes of textures. The pixels are grouped into texels. The objective is to identify a set of features that describe the texture of each texel and can be used as a basis for grouping the texels into classes with similar textures.

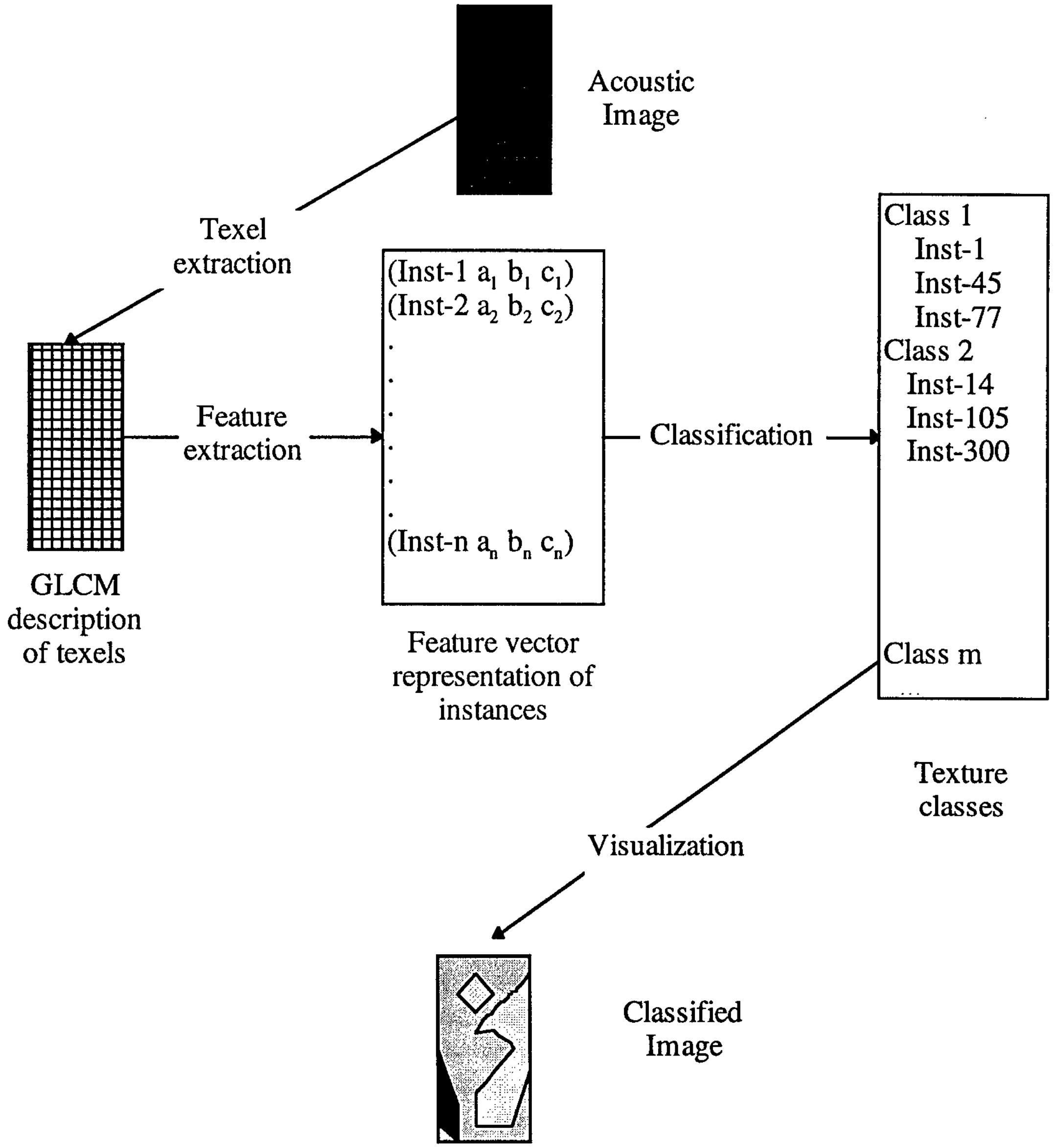

Figure 1. The Knowledge Discovery Process for Provincing of the Ocean Floor

We are currently working with only one data type - large acoustic images collected from a 100 kHz Chirp Side-Scan Sonar using a Data Sonics SIS1000. We are extracting texture information from the acoustic images and using machine learning techniques to identify classes of textures. We group the pixels of an image into texels, then group the texels into classes with similar textures. We are using the method described by Reed and Hussong (1989) to extract the features for each texel. In this approach, one computes a gray-level co-occurrence matrix (GLCM) for each texel at four different angles. One then computes texture features from the GLCMs. The texture features we are currently using are: angular second moment, contrast, entropy, angular inverse difference moment, and correlation. The feature vector for a texel consists of the mean texel intensity, the standard deviation of the texel intensity, and the five texture features computed in four different directions (a total of 22 features). We have been primarily using a Bayesian classifier called AutoClass (Cheeseman et al. 1988) to cluster the texels. In addition, we have just begun using a decision tree classifier called Cobweb (Fisher et al. 1993). The visualization of acoustic images and the classes that are identified is accomplished either by using the original UNISIPS software provided by NAVOCEANO, or by converting the images to ".pbm" or ".ppm" format and using the *xv* tool to view the images.

### 3.1 Texel Extraction with Fixed-Size Texels

The acoustic image data files that NAVOCEANO has made available to us are in UNISIPS format. In these gray scale images, pixel values are represented by integers ranging from 0 to 255. Geologists use the visual texture of the images to identify provinces of the ocean bottom. Visual texture is a measure of the variation of the gray scale values of a region of an image. Since texture is an attribute of groups of adjacent pixels, it is useful to group pixels into regions called texels and to extract features that describe the texture of the texel. A two-dimensional measure of texture that has proven useful is the gray-level co-occurrence matrix (GLCM) (Reed and Hussong 1989). The GLCM provides a measure of the frequency of finding different gray level values within a fixed distance and orientation from one another. This secondary matrix is used to calculate second order texture statistics. Following the notation of Reed and Hussong (1989), let F(x,y) represent a digital image with dimensions $L_x$ pixels by $L_y$ pixels where each pixel is quantized to $N_g$ gray levels. Each GLCM is an $N_g$ x $N_g$ square matrix, and each entry of the GLCM $S(i,j;\Theta,d)$ represents "the number of times there occurs in the image a pixel of intensity *i* neighbored by a pixel of intensity *j* in direction $\Theta$, at distance *d*" (Reed and Hussong 1989).

Texel size is a parameter for the algorithm, and the "best" texel size is determined empirically. Experiments that we have conducted using different texel sizes are described in the Experiments and Results section. For each texel, we have created four different GLCMs using $\Theta$ values of 0°, 45°, 90°, and 135° and a *d* value of 1. Reed and Hussong (1989) state that texels should contain at least $N_g^2$ pixels where $N_g$ is the number of gray levels to which the image has been quantized. If an $N_g$ value of 256 is used, each GLCM is very large and the texels become so large that the probability of having a mixture of textures in one texel is quite large. Scaling the number of gray levels to a smaller number allows one to use smaller texels and reduces the storage and computation requirement for the GLCM matrices. In our experiments, we have re-quantized the gray levels to ranges 0-15 or 0-31.

Each entry in the GLCM is determined by computing the number of adjacent ($d$ = 1) pixels of intensities *I* and *j*. Figure 2 shows an example of an image texel and the corresponding GLCM where the gray level of pixels has been quantized to 8 values (0..7), $\Theta = 0°$, and $d = 1$.

**TEXEL**

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 3 | 4 | 0 | 7 |
| 1 | 0 | 6 | 5 | 2 | 6 |
| 1 | 4 | 3 | 5 | 2 | 4 |
| 3 | 3 | 3 | 5 | 4 | 7 |
| 3 | 5 | 5 | 4 | 5 | 4 |

**GLCM**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 4 | 2 | 4 | 0 | 0 |
| 1 | 1 | 1 | 2 | 0 | 4 | 0 | 1 |
| 0 | 0 | 1 | 4 | 4 | 2 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |

Figure 2. An example of a 5 x 6 texel and the corresponding GLCM for $N_G = 8$, $\Theta = 0°$, and d=1.

Note that texel[0,0] and texel[0,1] are the only instances of adjacent 0 values, so GLCM[0,0] captures the fact that the zero value in texel[0,0] is next to the 0 value in texel[0,1], and that the 0 value in texel[0,1] is next to the 0 value in texel[0,0]. Any 0's in the GLCM represent pairs of values that never occur side by side. For example, GLCM[0,2] (which is 0) indicates that pixel values of 0 and 2 are never adjacent (left and right, since $\Theta = 0°$) in the texel. Also note that the diagonal ([0,0] to [7,7]) will contain only even numbers, and the upper right triangular portion of the GLCM is a mirror image of the lower left triangular portion.

### 3.2 Feature Extraction

Once the GLCM for a texel has been computed, it can be used to compute texture statistics that can serve as features for the classification system. Equations based upon those presented in Reed and Hussong (1989) were used to calculate the features. The equations are presented below:

**Normalization Factor**

$$R_0 = 2N_y(N_x - 1)$$

**Angular Second Moment**

$$ASM = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\left(\frac{S(i,j)}{R}\right)^2$$

**Contrast**

$$CON = \sum_{n=0}^{N_g-1} n^2 \left\{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{S(i,j)}{R}\right\}$$

$$|i-j| = n$$

**Entropy**

$$ENT = -\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{S(i,j)}{R} * \log(\frac{S(i,j)}{R})$$

**Angular Inverse Difference Moment**

$$AIDM = \sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{S(i,j)}{((1+(i-j)^2)*R)}$$

**Correlation**

$$COR = \frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\frac{(i*j)*S(i,j)}{R-\mu_x\mu_y}}{\sigma_x\sigma_y}$$

where $\mu_x$ and $\mu_y$ represent the means of the rows and columns, respectively, and $\sigma_x$ and $\sigma_y$ are the standard deviations of the rows and columns, respectively. Software has been developed to compute these feature values and write them to files suitable for use by AutoClass. The mean and standard deviation of the intensity value of the unquantized pixel values for each texel are also available as features.

### 3.3 Classification

The goal of classification is to assign the pixels (or texels) of an image to an "appropriate" class in such a way that some error measure is minimized. There are two basic approaches to accomplishing this classification, supervised learning and unsupervised learning. In supervised learning, a set of classes is identified by a domain expert and a subset of data instances are classified into the appropriate classes by the expert. The classification system is then trained with the preclassified instances (training set) and evaluated with other instances (test set). In unsupervised learning, the system must learn the classes as well as how to assign instances to classes. Unsupervised learning (also commonly called clustering) has the potential of uncovering previously unknown classes in the data. Cheeseman and Stutz (1996) have found that "discovery of previously unknown structure occurs most frequently when there are many relevant attributes describing each case, because humans are poor at seeing structure in a large number of dimensions."

In our first set of classification experiments, we used AutoClass (Cheeseman and Stutz 1996), an unsupervised Bayesian classification system. An input file in a format appropriate for AutoClass was constructed where each instance corresponded to a texel in the image and the feature values are those described above. AutoClass also allows one to specify a number of parameters for search such as the maximum duration of the search, the maximum number of tries, or the number of classes to be found. Once the classification is complete, AutoClass generates a report of the results, which can then be used in the visualization process. In our first set of experiments, we used all texels to train the classifier. This classification process often requires 6 to 10 hours to complete.

In an attempt to reduce the time required for classification, we have conducted a set of experiments in which a subset of the texel instances from an image are used to train the classifier and then the class membership of the remaining texels is determined by AutoClass in prediction mode. In this set of experiments, the training set was constructed by randomly selecting 10 percent of the texel data for an image and using the remaining 90% as the test set. Although some information may be lost in the prediction mode, the time difference for classification is extremely significant. One benchmark test that was run found that the prediction method took approximately 1/60 of the time required for a full search.

AutoClass generates an output file which contains all of the class information necessary for reconstructing an image from the classification data.

### 3.4 Visualization of the Results

Visualization of the results is accomplished by mapping each of the original texels to a cluster (produced in the classification step) and giving each pixel in this texel a color associated with this cluster. The resulting image is in the same format as the original image, so the same software (UNISIPS) can view it as the original image. This mapping process consists of two steps. The first step is to reorganize the data produced by the classification step while accumulating statistics to determine the color appropriate for each cluster. Then an image is built that is a duplicate of the original but with the pixel color chosen from: a) the original, if the pixel is not within a texel, b) the color assigned to the cluster the texel is associated with, if the pixel falls within a texel. Colors are assigned to the clusters by calculating the average pixel value for each cluster, sorting these clusters by this average, and assigning gray levels in ascending values equally distributed over the range 0..255. Figure 3 shows an example of an acoustic image and the corresponding image of classified texels from one experiment. The lightest areas in the classified image correspond to sand waves in the acoustic image.

### 3.5 Variable Size Texels Based on Region Growing

Texel extraction does carry with it some difficulties. As mentioned above, the technique involves "dividing the image into fixed rectangular regions of pixels;" that is, imposing a fixed grid over a naturally irregular seafloor map. Care must be taken in choosing the size of this grid: if the texels are too small (i.e., contain fewer pixels), there is a risk that they will not contain enough real textural information to be useful; if they are too large, it is more likely that individual texels will cover areas of varying texture, forming what are called "mixels" which are overloaded with inconsistent information. A technique called region growing is an attempt to use both of these extremes to an advantage. A grid of fixed-size rectangular units is still applied to the image, but in this case they are called *cells*, and they are much smaller than ordinary texels. However, these cells themselves are not the final product of this technique, being too small. Rather, they are analyzed, compared with each other, and joined to form larger, more texturally homogeneous and naturally shaped pixel sets called *regions.* These regions are the output analogous to the texels discussed above. A set of pixel statistics - mean, variance, and standard deviation - as well as GLCMs on four orientations (0, 45, 90, 135) are generated as they were for texels and fed to the classification step. The resulting class data can then be used to rebuild a clustered image, though this can take a little more effort than was required for texel visualization.

Individual cells are first submitted to a boundary test to see if they are fit to join a region. This boundary test is simply the comparison of the ratio of the call pixels' standard deviation to their mean against a user chosen limit - a threshold. A high ratio means less homogeneity in the cell pixels, indicating that the cell has pixels from more than one distinct region. If the ratio is higher than the threshold, the cell fails the test and is marked as a boundary, not participating in the region growing. Cells passing this test are then compared to neighboring cells that have already been processed. In this case, cells are analyzed a row at a time (with one previous row kept on hand) from left to right within a row, so that the cells available for comparison are those in the preceding row, northwest, north, and northeast and the one immediately to the left, or west. More accurately, the mean of the current cell being examined is compared with the mean of the parent regions of the neighboring cells. The current cell then joins the region whose difference is least and also falls within the limit of another different user chosen threshold value. As cells join, their pixel values are added into the region's overall statistics, and their interactions with other pixels in the region are recorded in the region's GLCMs, analogous to a texel's GLCMs. If none of the neighbor comparisons are within the limit, the cell is different enough to start its own new region (again along with new GLCMs).

This process is iterated over the entire acoustic image to produce several naturally shaped regions. Note that it is possible for a larger region to completely surround a smaller one - obviously the smaller one was still different enough from the larger to be created in the first place. Such a configuration can vary as the aforementioned threshold parameters are varied - the principle way for the user to interact with the region growing process.

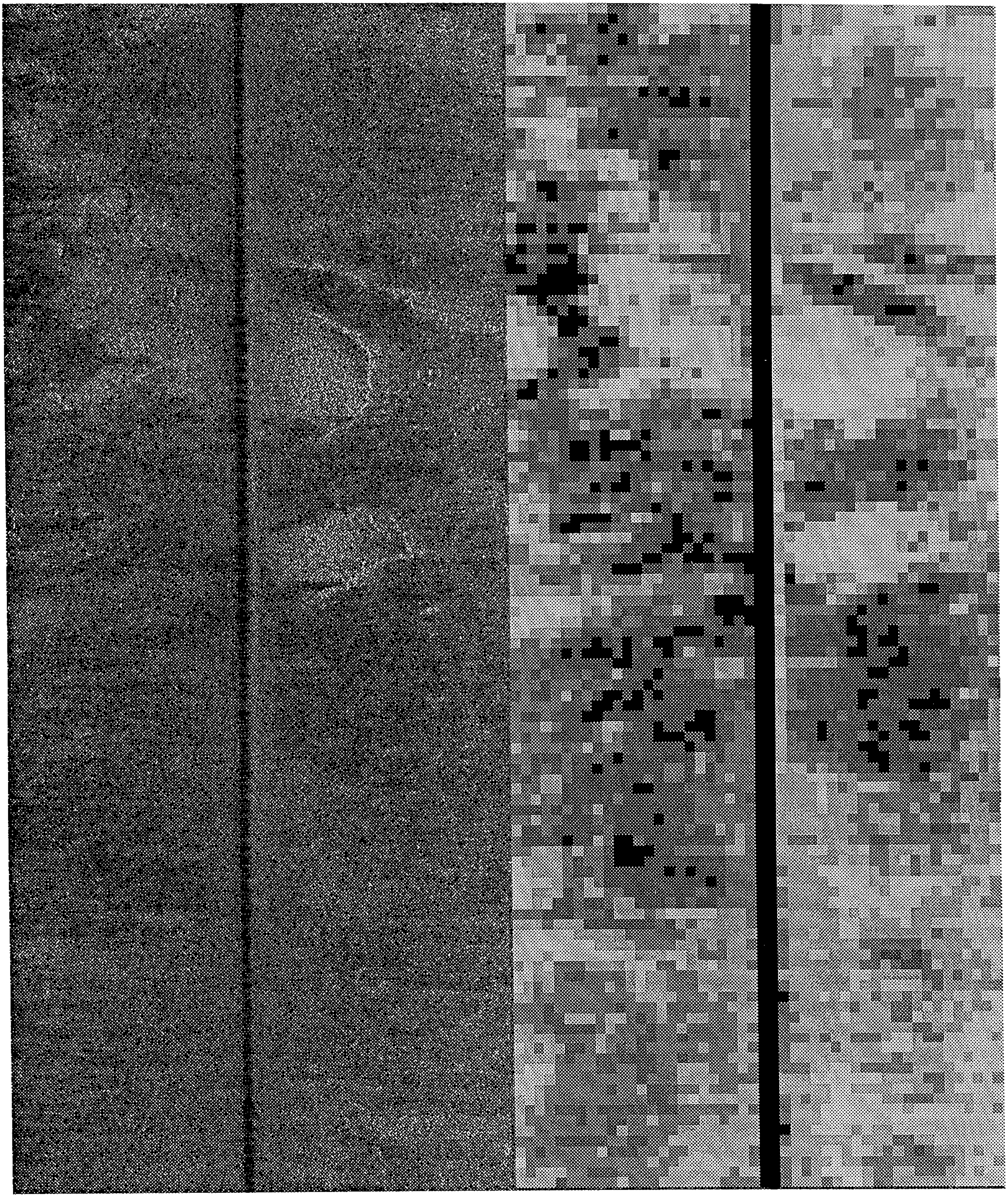

Figure 3. Acoustic image and corresponding texture classification.

Lowering the boundary threshold allows fewer cells to be eligible for region formation, demanding more homogeneity in cell pixels, while higher values will be more lax. Lowering the region threshold makes it more difficult for any given cell to join a particular region, demanding that a cell be very similar to a region to join it. This should result in a greater number of smaller, more homogeneous regions. Raising this threshold, on the other hand, should make it easier for cells to join a region with only some similarity required. This should result in fewer, larger regions that can vary somewhat with regard to internal textures. The precise effects of these settings with respect to clustering is still under investigation, Either way, one of the bonuses of the region growing technique is that it generally results in fewer samples, in this case, regions, than its texel counterpart, while still keeping with the same sample format, containing both statistics and GLCM data. This means less data overall for the classifier to deal with, which should make for shorter runs.

Reconstructing a new image for evaluation out of the clustered data is also somewhat similar to the texel version, but a lot more bookkeeping is involved. Much data regarding regions, cells, pixels, and test cases has to be kept on hand for the subsequent rebuilding. While with the texel method, each test case submitted to the classifier is a single texel, corresponding to a regular fixed box of pixels, in region growing test cases are whole regions which are made up of a variable number of cells in various locations in the original image. Getting from these regions to their constituent pixels is where the additional work is needed. Information on which cells belong to which regions is captured during the actual region growing stage and put aside. After classification, this data is used to map the region/class data back onto specific cells, and thus precise pixels.

In contrast to texel visualization, a more simplistic color mapping scheme has been adopted. The number of resulting classes from the clustering step is divided into the total possible gray values, in this case 256, to acquire the increment level between gray shades. The lowest class ends up with the lowest (darkest) gray and the highest class, the highest (lightest) gray regardless of the average gray level of the original pixels in the particular region in question. So, for example, if 20 classes were found, the whole number gray increment would be 256/20 = 12, class #1 would be displayed as a gray of 12 (rather dark), and class #20 would get a ray of 240 (very light). While this may produce images with colors that look very little like the original acoustic image, the textural features are quite discernible. The potential topsy-turvy allocation of the colors may even provide some beneficial contrast in some cases. However, future versions of the visualization may attempt to incorporate more accurate color matching. In addition, there is an alternate advanced version of the reconstruction software that can add bright red lines along the borders between regions, to further illuminate the different areas that region growing assembled.

## 4. Experiments and Results

To ensure that the individual feature equations were valid, we conducted a set of experiments in which the texels of an image were classified on each of the features individually. In these experiments, we set the AutoClass parameters to allow the classifier to run for 10 hours with an undetermined number of classes. The results indicated that the standard deviation (STDEV) calculation made during the GLCM process was invalid and that the equation for contrast (CON) was incorrectly implemented. Corrections were made to these calculations.

In addition to allowing us to identify feature equations that had been incorrectly implemented, the first set of experiments also allowed us to determine which topographical features were identified by the various attributes of the computed features. This is information that we will use in the definition of the object-oriented database.

Unlike the pictures resulting from experimental runs with other individual features, the picture generated by clustering on the angular inverse difference moment (AIDM) failed to extract many of the major topographical features of the original image. The arithmetic mean (MEAN) and STDEV produced the smallest number of classes. While this may indicate that the MEAN and STDEV are insignificant for a larger scale experiment that is attempting to recognize multiple topographical features, it is also the case that the MEAN and STDEV are most helpful when the objective is to keep the number of classes generated to a minimum.

In the second set of experiments, we defined AutoClass runs of 10 hours (with an undetermined number of classes for each run) that combined every possible pair of features (i.e., 21 pairs) to see how the features interacted with each other. The most interesting result from this set of experiments was that the images produced from the combination of AIDM and MEAN were relatively smooth and, at the same time, drew the most attention to the topographical features of the original image.

The third set of experiments was designed to determine the optimal texel size. For the first two sets of experiments, we used a texel size of 12 x 24 based on the experiments described by Reed and Hussong (1989). However, many of the resulting images had a large number of classes, too many to be useful in determining patterns within an image. What is desirable is to have a small number of classes that encompass similar texels. Increasing the texel size reduces the number of texels while increasing their size. We were hoping for a reduction in the number of classes without diluting the resulting images until they were no longer useful. We tried texel sizes of 14 x 14, 15 x 15, 16 x 16, 17 x 17, 18 x 18, 22 x 22, 26 x 26, and 52 x 52. We again set AutoClass for 10-hour runs and an undetermined number of classes. In this set of experiments, we had AutoClass classify based upon all of the attributes except MEAN and STDEV.

At this point, we attempted to limit the number of classes by using AutoClass's starting categorization value. AutoClass bases this value on either a predetermined value provided by the user or on the previous iteration. We defined a starting value of 20 for the first 20 attempts. This did not produce a smaller number of classes. However, once AutoClass went beyond the twentieth attempt, it began to find a larger number of classes than was set in the parameters. Of all the images resulting from these experiments, the 26 x 26 image seemed to offer the best compromise between texel size and area covered.

When AutoClass was set to run for 10 hours, the number of classes and the number of attempts to form classes would fluctuate depending on the complexity of the calculations involved in the combination of attributes. Since some limit must be set to eventually stop AutoClass, we defined the next set of experiments to limit the number of attempts. For instance, with a 52 x 52 texel size, since there were fewer texels to categorize and the calculations did not take as long as with more (smaller) texels, more calculations could be performed during 10 hours. As a result, the number of classes generated tended to increase. To have a more valid comparison between images, it was necessary to keep the number of attempts the same. Of the different experiments, the 14 x 14 texel size produced the smallest number of attempts since there were more texels in the image. The number of attempts from this experiment (26) was used for the basis of the next set of experiments.

The experiments with texel sizes were performed again with 26 x 26 and 52 x 52 texel sizes to determine the results of limiting the number of attempts. As before, we limited the number of classes during the first 20 attempts. Of the resulting images, the 26 x 26 texel size still seemed to provide the most useful information.

In the next set of experiments, we explored the number of classes determined by AutoClass. An AutoClass run can be set up not only with a limited number of classes for a certain number of attempts, but also for a set number of classes. In these experiments, we used the 26 x 26 texel size and all attributes except MEAN and STDEV. In different runs, we set the number of classes at 5, 10, 15, and 20. We limited AutoClass to 26 attempts (based on the previous set of experiments). The results showed that the limit of 20 classes produced the most reasonable results. Anything less tended to produce a less smooth image with the classes intermixed to an unacceptable degree.

The remaining experimental runs were all limited to 26 attempts and 20 classes. We tried to decrease the amount of time required for the experiments. Running a 26 x 26 image with all attributes except MEAN and STDEV, 20 classes, and 26 attempts required just over two hours to run. We took advantage of an AutoClass prediction mode that takes a small sample of the data values, classifies them, and predicts the classes for the remaining portion of the data. We redesigned the programs to select a random sample of 10% of the data, classify this data, then predict the classes for the entire data set based on the random sample. The results produced were comparable to using the entire data sets, and the run time was just under six minutes. Based on this, it appears desirable to use AutoClass in prediction mode in all future experiments that are abased on fixed-size texels.

## 5. Summary

In this report, we have described the first-year efforts in developing an oceanographic knowledge discovery system that can aid oceanographers in the identification of certain oceanographic features by the analysis of a variety of data such as acoustic imagery, grid data, model output, and sensor data. During the first year, we focused our attention primarily on the development of a discovery system that can identify particular oceanographic features (in our case, sand waves) by applying machine learning techniques to acoustic imagery data. We have conducted a series of experiments to determine the texel sizes, data attributes, and number of classes that produce the most useful images for recognizing the features. The results of those experiments have been reported here.

## References

Brachman, Ronald, and Tej Anand. 1996. The process of knowledge discovery in databases: A human-centered approach. *Advances in knowledge discovery and data mining.* Edited by Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. Menlo Park, CA: AAAI Press. 37-58.

Chan, Philip K., and Salvatore J. Stolfo. 1996. Sharing learned models among remote database partitions by local meta-learning. In *Proceedings of the second international conference on knowledge discovery and data mining, August 2-4, 1996, Portland Oregon.* Edited by Evangelos Simoudis, Jiawei Han and Usama Fayyad. Menlo Park, CA: AAAI Press. 2-7.

Chen, M.-S., J. Han, and P. S. Yu. 1996. Datamining: An overview from a database perspective. *IEEE Transactions on Knowledge and Data Engineering* 8(6) 866-883.

Cheeseman, Peter, and John Stutz. 1996. Bayesian classification (AutoClass): Theory and results. *Advances in knowledge discovery and data mining.* Edited by Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. Menlo Park, CA: AAAI Press. 158-180.

Cheeseman, P. J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman. 1988. AutoClass: A Bayesian classification system. In *Proceedings of the fifth international conference on machine learning*. Reprinted in *Readings in machine learning*, edited by Jude W. Shavlik and Thomas G. Dietterich, San Mateo, CA: Morgan Kaufmann Publishers, Inc., 296-306.

Cheung, D. W. , V. T. Ng, A. W. Fu, and Y. Fu. 1996. Efficient mining of association rules in distributed databases. *IEEE Transactions on Knowledge and Data Engineering* 8(6) 911-922.

Fayyad, Usama, David Haussler, and Paul Stolorz. 1996. KDD for science data analysis: Issues and examples. In *Proceedings of the second international conference on knowledge discovery and data mining, August 2-4, 1996, Portland Oregon.* Edited by Evangelos Simoudis, Jiawei Han and Usama Fayyad. Menlo Park, CA: AAAI Press. 50-56.

Fayyad, Usama M., Gregory Piatetsky-Shapiro, and Padhraic Smyth. 1996. From data mining to knowledge discovery: An overview. *Advances in knowledge discovery and data mining.* Edited by Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. Menlo Park, CA: AAAI Press. 1-36.

Fisher, Doug, Lin Xu, James R. Carnes, Yoran Reich, Steven J. Fenves, Jason Chen, Richard Shiavi, Gautam Biswas, and Jerry Weinberg. 1993. Applying AI clustering to engineering tasks. *IEEE Expert* 8(6): 51-60.

Reed, Thomas Beckett IV, and Donald Hussong. 1989. Digital image processing techniques for enhancement and classification of SeaMARC II side scan sonar imagery. *Journal of Geophysical Research* 94(B6): 7469-90.

Shek, Eddie C., Richard R. Muntz, Edmund Mesrobian, and Kenneth Ng. 1996. Scalable exploratory data mining of distributed geoscientific data. In *Proceedings of the second international conference on knowledge discovery and data mining, August 2-4, 1996, Portland Oregon.* Edited by Evangelos Simoudis, Jiawei Han and Usama Fayyad. Menlo Park, CA: AAAI Press. 32-37.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
Sept. 30, 1997

3. REPORT TYPE AND DATES COVERED
Performance Report, 9/1/96 - 8/31/97

4. TITLE AND SUBTITLE
Knowledge Discovery in an Object-Oriented Oceanographic Database System

5. FUNDING NUMBERS
Grant No. N00014-96-1-1276
PR No. 96PR07924-00

6. AUTHOR(S)
Julia Hodges
Susan Bridges

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(ES)
Mississippi State University
P.O. Box 6156
Mississippi State, MS 39762

8. PERFORMING ORGANIZATION REPORT NUMBER
P.O. Code 321SI

9. SPONSORING / MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
Office of Naval Research Regional Office Atlanta
101 Marietta Tower Suite 2805
101 Marietta St.
Atlanta, GA 30321-0008

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AGO Code N66020

11. SUPPLEMENTARY NOTES

a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release

12. DISTRIBUTION CODE
N68892

13. ABSTRACT *(Maximum 200 words)*

The rate at which scientific data is collected today has overwhelmed the ability of scientists to store and analyze the data. Current research in knowledge discovery in databases is addressing this problem by developing techniques that can consider large quantities of data and automatically identify information that is of interest in a particular problem domain.

This report describes the results of the first year's efforts in the development of a knowledge discovery system for use by oceanographers at the Naval Oceanographic Office at the Stennis Space Center in the identification of certain oceanographic features. The system conssits of two major components: an object-oriented oceanographic database that can support the retrieval of data along various parameters of interest (such as a certain geographic area of a certain date) and a discovery system that can identify the features of interest. During the first year of this project, we (in consultation with the scientists at the Stennis Space Center) named the identification of sand waves in acoustic imagery as the first task to be addressed by our system.

14. SUBJECT TERMS
Knowledge Discovery, Data Mining, Acoustic Imagery, Object-Oriented Database

15. NUMBER OF PAGES
12

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
unclassified

20. LIMITATION OF ABSTRACT